Astronomical Data Analysis Software and Systems VIII
ASP Conference Series, Vol. 172, 1999
D. M. Mehringer, R. L. Plante, and D. A. Roberts, eds.

Pipeline Processing of Data in the VLT Era

Preben Grosbøl, Klaus Banse, Pascal Ballester

European Southern Observatory,
Karl-Schwarzschild-Str. 2, D-85748 Garching, Germany

Abstract. The VLT Data Flow System pipeline and quality control subsystem provides a general infrastructure for standard reduction and quality assessment of data obtained at the VLT facility. The main design challenge is to support a wide range of instruments in a distributed environment. The pipeline system can be configured, through a set of ASCII files, to handle several instrument simultaneously. It was designed using the object-oriented methodology and major parts of the baseline version will be written in Java using OMG/CORBA technology to support distributed objects.

1. Introduction

The VLT Data Flow System (DFS) provides a single, homogeneous, end-to-end system for handling science data from the VLT facility (see Grosbøl & Peron 1997). It can be divided into three main parts, namely: a) pre–observation tasks which include preparation of observing proposals, detailed specification of observations and tentative scheduling, b) observation support containing archiving services, and c) post–observation processing including pipeline reduction and quality control of data acquired.

This paper focuses on the post-observation modules with emphasis on the standard pipeline reduction of VLT data. A discussion of the quality control aspects was given by Ballester et al. (1998).

2. Requirements and Challenges

The vast amount of data produced by the VLT and the multitude of instruments demand that raw data can be reduced very efficiently and with a minimum of manual intervention. The DFS pipeline has been designed for this purpose and will be used in four main scenarios:

- near real time reduction of data at the VLT observatory to provide a first assessment of data quality,
- off-line reduction of data at the ESO headquarters to offer users standard reduced data products for service-mode observations,
- data reduction at the users home institute to make a more customized processing possible, and
- re–processing of data from the VLT Science Archive.

Whereas it is trivial to make an explicit pipeline to reduce data from a given instrument, the main challenge for the DFS pipeline is to create a unique infrastructure which can serve all of the more than 15 different instruments on the four 8.2 m telescopes plus ancillary units for interferometry and wide field imaging. The long expected life time of the facility makes it mandatory to rely on a single concept to ease operation and reduce maintenance costs while employing a modular design and thereby enable a gradual replacement of components in the course of time.

There must be a clear separation between the pipeline infrastructure and data processing tasks to ensure that any suitable data reduction system (DRS) can be used. Some reduction tasks for a specific instrument may already be available in a particular system (e.g., AIPS++ could provide interferometric reduction procedures). It is also prudent to assume that not all current DRS's will be fully updated or supported over the next decades. A smooth migration of reduction tasks from one to another system would therefore be important. Further, the large data volume also makes it essential to support parallel processing to take advantage of multi–processor or loosely coupled computers e.g. Beowulf type of systems.

3. Pipeline Model and Assumptions

The pipeline processing model assumes that all raw frames can be uniquely associated to a specific instrument and the full description of its setup can be obtained from their FITS headers. It must also be possible to determine the observational and operational context of a raw frame relative to others by analyzing its FITS keywords. This provides a hierarchical grouping of frames e.g. based on their relation to Observation Blocks and Templates (see Grosbøl & Peron 1997). Each instrument must define a unique classification of all frames used by the pipeline including raw frames, calibration data and products generated.

Actual pipeline tasks are triggered by different events such as *arrival of new raw frame* or *end of observing template*. The frames associated to the event are identified and the necessary set of reduction recipes are obtained following rules defined for the instrument. The recipes specify the algorithms and the required parameter including calibration data such as CCD flat fields or wavelength tables. Calibration files are either stored in a local database or file directory structure. The appropriate calibration frames for a given input data set are found through their classification and by matching a primary key defined as a set of FITS keywords.

4. Architectural Design

The architecture of the pipeline system is based on a distributed, object oriented design. The system can either be driven by events or activated through graphical user interfaces. Five main applications are defined to support the basic pipeline functionality:

- **Pipeline:** is responsible for the standard reduction of data. It creates the explicit reduction tasks (*ReductionBlocks*) to be executed following a given event.

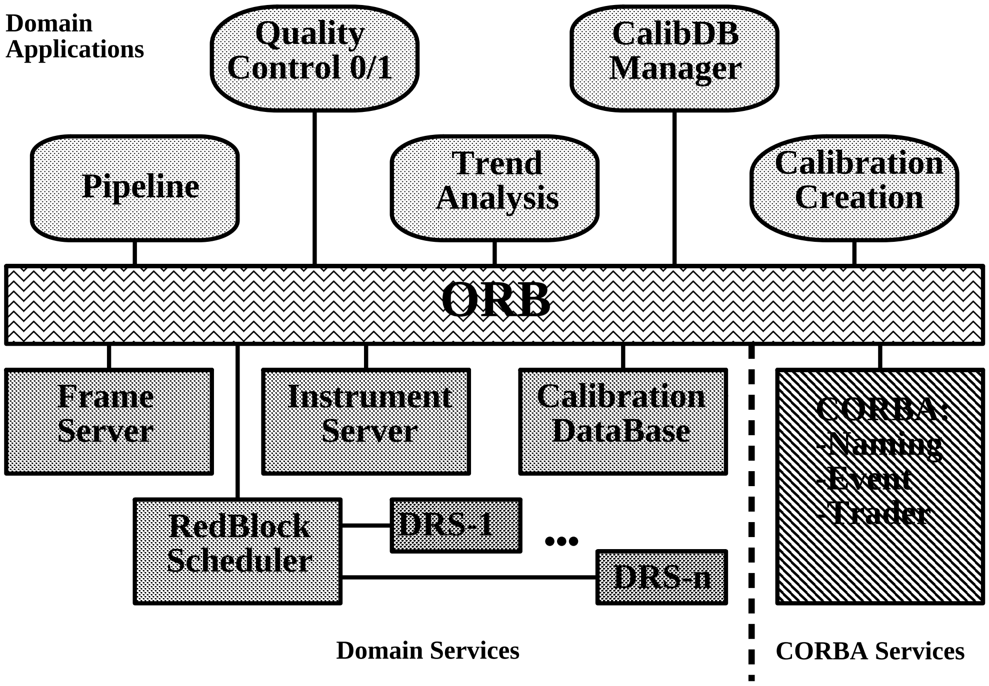

Figure 1. Main applications and services of the DFS Pipeline subsystem.

- **Quality Control:** checks the quality of both raw and reduced data by comparing them with requirements and models.
- **Trend Analysis:** monitors the calibration solutions to identify possible variations.
- **Calibration Database Manager:** controls and maintains the calibration data used for the pipeline reductions.
- **Calibration Creation:** creates new calibration data which after a certification process may be included in the database.

The first implementation is aimed mainly at an automatic, batch type environment but it is expected that more clients will be added to provide better interactive control. The applications use a set of general services:

- **Frame Server:** determines context and classification of new frames and groups them.
- **Instrument Server:** provides several instrument specific facilities e.g. FITS keyword mapping, classification, reduction recipe descriptions and rules for required reduction steps.
- **Calibration Database:** is the depository for calibration data and support search methods to locate them.
- **ReductionBlock Scheduler:** receives *ReductionBlocks* and schedules them for execution by one of the Data Reductions Systems it has access to.

The communication between clients and servers is based on the OMG/CORBA distributed object model. The components are shown in Fig. 1 together with several CORBA services which may be used. Whereas the Naming Service is available in most CORBA implementations, the Event and Trader Services are not yet standardly provided.

5. Configuration and Implementation

It is essential that new instruments can be easily integrated into the pipeline environment. This is facilitated by defining their behaviors in a set of ASCII configuration files. They make it possible to define the following instrument specific items:

- **FITS Keyword Mapping:** specifies the relation between the FITS keywords and information used by the pipeline.
- **Frame Classification:** defines the criterion for classifying individual frames based on a boolean expression. It also gives the structure of the frame and the primary key used for the association of calibration data.
- **Reduction Recipe Specification:** lists the available recipes with their formal parameters including calibration data and default values.
- **Reduction Rule:** defines the list of action or recipes to be executed depending on the context and event type.

The files are under configuration control and make it possible to process data from the Science Archive using the appropriate versions for the instrument definitions.

It is foreseen that the DFS pipeline will be employed at computer systems distributed over the ESO sites and possibly exported to external institutes. Web based interfaces are also expected to play an important role in operating and monitoring the pipelines. The major parts of the DFS pipeline will be implemented in Java which provides excellent support for distributed object systems and user interfaces. It is expected that OMG/CORBA based tools will be used for the object bus and general services.

References

Grosbøl, P. & Peron, M. 1997, in ASP Conf. Ser., Vol. 125, Astronomical Data Analysis Software and Systems VI, ed. G. Hunt & H. E. Payne (San Francisco: ASP), 23

Ballester, P., Kalicharan, V., Banse, K., Grosbøl, P., Peron, M., & Wiedmer, M. 1998, in ASP Conf. Ser., Vol. 145, Astronomical Data Analysis Software and Systems VII, ed. R. Albrecht, R. N. Hook, & H. A. Bushouse (San Francisco: ASP), 259